

Kartendarstellungen mit matplotlib Toolkit basemap

Simon von Hall

9. Dezember 2014

Inhaltsverzeichnis

1	Motivation	4
2	Kartendarstellungen	4
2.1	Azimuthale äquidistante Projektion	4
2.2	Gnomonische Projektion	5
2.3	Orthographische Projektion	5
2.4	Geostationary Projection	5
2.5	Near-Sided perspektivische Projektion	5
2.6	Mollweide Projection	5
2.7	Hammer Projektion	6
2.8	Robinson Projektion	6
2.9	Eckert 4 Projektion	6
2.10	Kavrayskiy 7 Projektion	6
2.11	McBryde-Thomas Projektion	7
2.12	Sinusoidale Projektion	7
2.13	Äquidistante Zylinder Projektion	7
2.14	Cassini Projektion	8
2.15	Mercatorprojektion	8
2.16	Transversale Mercatorprojektion	8
2.17	Schiefe Mercatorprojektion	8
2.18	Polykonische Projektion	8
2.19	Miller Zylinderprojektion	9
2.20	Stereographische Gall Projektion	9
2.21	Flächentreue Zylinderprojektion	9
2.22	Lambert Winkeltreue Projektion	9
2.23	Azimuthale Flächentreue Lambert-Projektion	10
2.24	Stereographische Projektion	10
2.25	Längentreue Kegelprojektion	10
2.26	Flächentreue Albert-Projektion	10
2.27	Polare stereographische Projektion	11
2.28	Polare azimutale Lambertprojektion	11
2.29	Polare azimuthale äquidistante Projektion	11
2.30	Van der Grinten Projektion	11
3	Basemap	13
3.1	Einführung	13
3.2	Abhängigkeiten von basemap	13
3.3	Erstellen einer Karte mit Basemap	13
3.4	Zeichnen der Karte	16
3.5	Bild auf eine Karte zeichnen	19
3.6	Daten plotten	20
3.6.1	Isobaren plotten	20

3.6.2	Windmarken plotten	20
3.6.3	Windvektoren plotten	20
3.6.4	gekrümmte Strecken plotten	21
3.6.5	Viele Punkte plotten	21
3.6.6	Linienzüge zeichnen	21

1 Vorwort

In dieser Seminararbeit möchte ich das Python Modul `Basemap` vorstellen. Dazu werde ich die verschiedenen Kartenprojektionen die es unterstützt kurz beschreiben. Danach gehe ich auf die Funktionen des Moduls ein und beschreibe was diese machen. Dabei gehe ich dann noch ganz kurz darauf ein was für Voraussetzungen das Modul erwartet. Ich erkläre außerdem kurz wie man einfach mit dem Koordinatensystem umgehen kann. Zum Schluss erkläre ich noch ganz kurz wie man die Karten die man erstellt hat speichert.

2 Kartendarstellungen

In diesem Kapitel möchte ich die vom Modul `Basemap` unterstützten Kartenprojektionen kurz beschreiben.

2.1 Azimuthale äquidistante Projektion

Bei dieser Projektion ist die kürzeste Entfernung vom Mittelpunkt der Karte zu einem beliebigen Anderen Punkt eine gerade Linie. Das bedeutet das alle Punkte die in auf einem Kreis um den Kartenmittelpunkt liegen, äquidistant sind. Nachteil:

Die Gebiete die auf der anderen Seite der Welt liegen werden sehr verzerrt dargestellt. Daher ist diese Projektion für Weltkarten eher ungeeignet.

2.2 Gnomonische Projektion

In der gnomonischen Projektion werden alle Längengrade als gerade Linien dargestellt. Das Besondere der gnomonischen Projektion ist, dass der Projektionspunkt im Mittelpunkt der Erde liegt. Da hier von Innen nach Außen projiziert wird, nimmt die Verzerrung mit der Entfernung vom Kartenmittelpunkt zu. Die gnomonische Projektion ist keine globale Projektion.

2.3 Orthographische Projektion

Bei der orthographischen Projektion wird die Erde aus einer unendlichen Entfernung abgebildet. Die Verzerrung ist an den Grenzen nicht übermäßig stark. Die orthographische Projektion ist keine globale Projektion. Durch die große Entfernung wirkt die Projektion dreidimensional.

2.4 Geostationary Projection

In der geostationären Projektion wird die Erde aus der Perspektive eines geostationären Satelliten. Vorteil:

- Wenn die Position des Satelliten bekannt ist, kann man dessen Bilder als Hintergrund verwenden (siehe ??)

Nachteil:

- Die andere Seite der Erde wird nicht dargestellt.
- Entfernungen zwischen 2 Punkten werden auf Kreisbögen gemessen.

2.5 Near-Sided perspektivische Projektion

Die Near Sided Perspective zeigt die Erde aus der Sicht eines Satelliten. Also ist es im Prinzip das selbe wie die geostationäre Projektion.

2.6 Mollweide Projection

Bei der Mollweiden Projektion wird die Erde als Oval dargestellt. Die Mollweiden Projektion ist flächentreu. Der Äquator und der Nullmeridian werden bei der Mollweiden Projektion maßstabsgetreu wieder gegeben. Breitenkreise werden bei der Mollweiden Projektion als Geraden dargestellt. Die Längengrade sind als Ellipsen dargestellt.

2.7 Hammer Projektion

Die Hammer Projektion ist wie die Mollweiden Projektion eine flächentreue Projektion. Bei der Hammer Projektion wird die Erde ebenfalls als Oval dargestellt. Allerdings werden die Breitenkreise im Gegensatz zur Mollweiden Projektion als Ellipsen dargestellt, dadurch ist die Verzerrung an den Rändern nicht so stark. Nachteil bei dieser Art der Darstellung ist, dass die Erde an den Polen gestaucht wird.

2.8 Robinson Projektion

Die Robinson Projektion ist eine globale Projektion. Die Erde wird hier annähernd Oval dargestellt. Die Pole werden allerdings in dieser Darstellung nicht abgedeckt. Breitenkreise werden in der Robinson Projektion als Geraden dargestellt. Bei dieser Darstellung wurden die Verzerrungen reduziert. Nachteil der Robinson Projektion ist das die Pole nicht erfasst werden.

2.9 Eckert 4 Projektion

Die Eckert 4 Projektion ist sehr ähnlich wie die Robinson Projektion, allerdings ist Sie flächentreu. Deshalb ist die Darstellung an den Polen gestaucht. Die Erde wird wie auf einem Reifen dargestellt. Die Seitenränder sind in dieser Projektion Halbkreise.

Formel:

$$\begin{aligned}\mathcal{X} &= \frac{2}{\sqrt{4\pi + \pi^2}} \mathcal{R}(\lambda - \lambda_0^1)(1 + \cos \theta) \\ \mathcal{Y} &= 2\sqrt{\frac{\pi}{4 + \pi}} \mathcal{R} \sin \theta \\ \theta + \sin \theta \cos \theta + 2 \sin \theta &= \left(2 + \frac{\pi}{2}\right) \sin \varphi\end{aligned}$$

2.10 Kavrayskiy 7 Projektion

Die Kavrayskiy 7 Projektion ist der Robinson Projektion sehr ähnlich. Sie stellt die Erde wieder annähernd oval dar. Die Breitenkreise werden in dieser Projektion als Geraden dargestellt. Diese Projektion stellt einen Kompromiss zwischen winkeltreuen und flächentreuen Projektionen dar. Die Pole sind in dieser Darstellung sehr breit gezogen.

Formel:

$$\mathcal{X} = \frac{3\lambda}{2\pi\sqrt{\frac{\pi^2}{3} - \varphi^2}} \quad (1)$$

$$\mathcal{Y} = \varphi \quad (2)$$

2.11 McBryde-Thomas Projektion

Diese Projektion ist eine flächentreue Darstellung der Erde. Die Breitenkreise werden als Geraden dargestellt. Die Längengrade werden als Bögen dargestellt. Dabei haben die Längengrade, auf einem Breitenkreis, immer den gleichen Abstand zueinander. Die Breitenkreise hingegen stehen immer näher je näher man den Polen kommt. Die Pole sind auf ein Drittel des Äquators gestreckt. Der Nullmeridian ist in dieser Projektion 0,45 mal so lang wie der Äquator.

2.12 Sinusoidale Projektion

Die sinusoidale Projektion ist eine flächentreue Projektion. Auch in dieser Projektion werden die Breitenkreise als Geraden dargestellt. Das besondere an der sinusoidalen Projektion ist das die Länge der Breitenkreise relational zu $\cos \varphi$ ist, dies führt zu einer starken Verzerrung außerhalb der Mitte. Vorteil der sinusoidalen Projektion:

- Die Projektion ist einfach zu berechnen.

Nachteil der sinusoidalen Projektion:

- Die Projektion ist nicht sehr anschaulich.

Formel:

$$\mathcal{X} = \mathcal{R} \cdot (\lambda - \lambda_r) \cdot \cos \varphi \quad (3)$$

$$\mathcal{Y} = \mathcal{R} \cdot \varphi \quad (4)$$

2.13 Äquidistante Zylinder Projektion

Die äquidistante Zylinder Projektion ist die einfachste Projektion. Sie stellt die Erde einfach in Längen- und Breitengrad dar. Dabei entsteht ein gleichmäßiges Gitterraster. Die Projektion ist weder winkel- noch flächentreu, das heißt, dass die Verzerrung mit der Entfernung vom Mittelpunkt der Karte zunimmt.

Vorteil der äquidistanten Zylinder Projektion:

- Die Projektion ist sehr einfach zu berechnen.

Nachteil der äquidistanten Zylinder Projektion:

- Die Verzerrungen wirken sowohl auf die Fläche als auch auf die Abstände aus.

Formel:

$$\mathcal{X} = \lambda \quad (5)$$

$$\mathcal{Y} = \varphi \quad (6)$$

2.14 Cassini Projektion

Bei der Cassini Projektion wird die Erde zuerst gedreht, sodass der Äquator zum Nullmeridian wird. Danach wird dann eine äquidistante Zylinder Projektion angewendet.

2.15 Mercatorprojektion

Die Mercatorprojektion ist eine winkeltreue Zylinderprojektion. Sie erreicht dabei nie Pole. Die Längengrade verlaufen in dieser Projektion parallel und haben den gleichen Abstand zueinander. Die Breitengrade sind ebenfalls parallel zueinander haben aber unterschiedliche Abstände. Die Verzerrung nimmt mit Polnähe zu, das heißt, dass die Karte zu den Polen hin gestreckt wird.

2.16 Transversale Mercatorprojektion

Bei der transversalen Mercatorprojektion wird der Globus zuerst um 90° gedreht, so dass der 0 Meridian zu Äquator wird. Danach wird eine normale Mercatorprojektion erstellt.

2.17 Schiefe Mercatorprojektion

Bei der schiefen Mercatorprojektion kann die zentrale Linie jede Linie zwischen zwei Punkten sein und nicht wie bei der Mercatorprojektion nur ein Breitengrad oder der transversen Mercatorprojektion nur ein Längengrad.

2.18 Polykonische Projektion

Die polykonische Projektion ist eine globale Projektion. Hierbei werden auf dem zentralen Meridian unendlich viele Kegel aufgebaut. Dabei entstehen

nicht konzentrische Breitenkreise. Die Projektion geht an den Polen auseinander. Die Verzerrung der Fläche nimmt mit der Entfernung vom zentralen Meridian der Karte zu. Die Winkel sind lokal entlang des zentralen Längengrades genau, sonst sind sie verzerrt.

2.19 Miller Zylinderprojektion

Die Miller Zylinderprojektion ist eine globale Projektion. Sie ist der Mercatorprojektion sehr ähnlich. Allerdings ist hier die Verzerrung an den Polen anders. Die Pole werden nicht mehr so stark gesteckt, dafür ist die Miller-Projektion nicht winkeltreu. Dafür reicht die Miller Zylinderprojektion bis an die Pole.

2.20 Stereographische Gall Projektion

Die stereographische Gall Projektion ist eine globale Zylinderprojektion. Die Gall Projektion hat zwei Standartparallelen bei 45°N und 45°S . Die Verzerrung der Fläche und Winkel nimmt mit Abstand zu den Standartparallelen zu. Die Verzerrung ist allgemein an den Polen sehr stark.

2.21 Flächentreue Zylinderprojektion

Die flächentreue Zylinderprojektion ist eine flächentreue Zylinderprojektion wie es der Name bereits sagt.

2.22 Lambert Winkeltreue Projektion

Die winkeltreue Lambert Projektion ist winkeltreu. Die Längengrade werden als Geraden dargestellt. Die winkeltreue Lambert Projektion ist eine Kegelprojektion.

Formel:

$$\begin{aligned}\mathcal{X} &= \rho \sin(n(\lambda - \lambda_0)) \\ \mathcal{Y} &= \rho_0 - \rho \cos(n(\lambda - \lambda_0)) \\ \rho &= F \cot^n\left(\frac{1}{4}\pi + \frac{1}{2}\varphi\right) \\ n &= \frac{\ln(\cos \varphi_1 \sec \varphi_2)}{\ln(\tan(\frac{1}{4}\pi + \frac{1}{2}\varphi_2) \cot(\frac{1}{4}\pi + \frac{1}{2}\varphi_1))} \\ F &= \frac{\cos \varphi_1 \tan^n(\frac{1}{4}\pi + \frac{1}{2}\varphi_1)}{n}\end{aligned}$$

2.23 Azimuthale Flächentreue Lambert-Projektion

Die flächentreue Lambert-Projektion ist eine globale Projektion. Diese Projektion wird um einen Punkt herum aufgebaut. Dabei bleibt der 3D Abstand jedes Punktes zum Mittelpunkt der Projektion erhalten. Das sorgt dafür, dass der Winkel mit zunehmender Entfernung vom Mittelpunkt stärker verzerrt werden. Diese Projektion stellt die Erde als Kreis dar.

2.24 Stereographische Projektion

Die stereographische Projektion ist eine winkeltreue Projektion. Sie ist allerdings nicht flächentreu. Man kann mit dieser Projektion die ganze Erde abbilden, allerdings nimmt die Verzerrung sehr schnell stark zu, weshalb die stereographische Projektion für globale Abbildungen eher ungeeignet ist. Diese Projektion erhält man, wenn man von einem Ausgangspunkt geraden durch jeden Punkt der Erde zieht, die Schnittpunkte dieser Geraden mit der Projektionsebene sind die Punkte auf die projiziert wird. Die Projektionsebene kann grundsätzlich frei positioniert werden.

2.25 Längentreue Kegelprojektion

Die längentreue Kegelprojektion ist eine globale Projektion. Sie ist weder winkel- noch flächentreu. Die Verzerrung nimmt zum Kartenrand hin zu. Die Breitenkreise sind in dieser Projektion gleichmäßig verteilt, und die Längengrade werden als Geraden dargestellt. Die Projektion bildet die Erde auf einen Kegelmantel ab.

2.26 Flächentreue Albert-Projektion

Die flächentreue Albert-Projektion ist eine globale Kegelprojektion. Sie bildet die Erde auf einen Kegelmantel ab. Sie benutzt für die Projektion 2 Standardbreitenkreise.

Formel:

$$\begin{aligned}
\mathcal{X} &= \rho \sin \theta \\
\mathcal{Y} &= \rho_0 - \rho \cos \theta \\
n &= \frac{1}{2}(\sin \varphi_1 + \sin \varphi_2) \\
\theta &= n(\lambda - \lambda_0) \\
\tau &= \cos^2 \varphi_1 + 2n \sin \varphi_1 \\
\rho &= \frac{\sqrt{\tau - 2n \sin \varphi}}{n} \\
\rho_0 &= \frac{\sqrt{\tau - 2n \sin \varphi_0}}{n}
\end{aligned}$$

2.27 Polare stereographische Projektion

Die polare stereographische Projektion ist eine stereographische Projektion die einen der beiden Pole als Kartenzentrum hat.

2.28 Polare azimutale Lambertprojektion

Die polare azimutale Lambertprojektion hat ebenfalls einfach nur einen Pol als Kartenzentrum.

2.29 Polare azimuthale äquidistante Projektion

Diese Projektion hat ebenfalls einfach einen Pol als Mittelpunkt.

2.30 Van der Grinten Projektion

Die van der Grinten Projektion ist eine globale Projektion, die die Erde auf einen Kreis projiziert. Diese Projektion ist weder winkel- noch flächentreu. Die Verzerrung nimmt zum Rand hin zu. Die van der Grinten Projektion ist um den Äquator zentriert.

Formel:

$$\begin{aligned}
\mathcal{X} &= \frac{\pm\pi(A(G-P^2) + \sqrt{A^2(G-P^2)^2 - (A^2+P^2)(G^2-P^2)})}{P^2+A^2} \\
\mathcal{Y} &= \frac{\pm\pi(PQ - A\sqrt{(A^2+1)(P^2+A^2) - Q^2})}{P^2+A^2} \\
A &= \frac{1}{2} \left| \frac{\pi}{\lambda - \lambda_0} - \frac{\lambda - \lambda_0}{\pi} \right| \\
G &= \frac{\cos \theta}{\sin \theta + \cos \theta - 1} \\
P &= G \left(\frac{2}{\sin \theta} - 1 \right) \\
\theta &= \arcsin \left| \frac{2\varphi}{\pi} \right| \\
Q &= A^2 + G \\
\text{Falls } \varphi = 0 : & \\
\mathcal{X} &= \lambda - \lambda_0 \\
\mathcal{Y} &= 0 \\
\text{Falls } \lambda = \lambda_0 \text{ oder } \varphi = \pm \frac{\pi}{2} : & \\
\mathcal{X} &= 0 \\
\mathcal{Y} &= \pm \pi \tan \frac{\theta}{2}
\end{aligned}$$

3 Basemap

3.1 Einführung

Das Toolkit *Basemap* ist ein Pythonmodul zur Bearbeitung von Karten. Es bietet viele verschiedene Arten Karten in 2D darzustellen (siehe Projektionen ??) Es ermöglicht einem auch das Plotten auf den Karten. Hierbei kann man, dann auch Längen- und Breitengrad als Positionsangabe nutzen. Das *Basemap* Toolkit wandelt die Koordinaten dann mit der *PROJ4* Library in die entsprechenden 2D Koordinaten um. Beim Plotten greift das Modul auf Funktionen des Moduls *pyplot* und *matplotlib* zurück.

3.2 Abhängigkeiten von basemap

Das Toolkit setzt folgende Voraussetzungen:

- Python
- Numpy
- Matplotlib
- GEOSlib (ist im Projekt enthalten)
- PROJ4 (ist im Projekt enthalten)
- dateutil
- pyparsing
- six
- libpng
- PIL (ist optional)
PIL wird nur gebraucht um auf mehr Bildformate zugreifen zu können.

3.3 Erstellen einer Karte mit Basemap

Eine Karte erzeugt man mit der Klasse *Basemap*. Die Funktion ist wie folgt definiert:

```
Basemap(llcrnrlon=None, llcrnrlat=None, urcrnrlon=None, urcrnrlat=None, llcrnrx=None, llcrnry=None, urcrnrx=None, urcrnry=None, width=None, height=None, projection='cyl', resolution='c', area_thresh=None, rsphere=6370997.0, ellps=None, lat_ts=None, lat_1=None, lat_2=None, lat_0=None, lon_0=None,
```

```
lon_1=None, lon_2=None, o_lon_p=None, o_lat_p=None, k_0=None,  
no_rot=False, suppress_ticks=True, satellite_height=35786000,  
boundinglat=None, fix_aspect=True, anchor='C', celestial=False,  
round=False, epsg=None, ax=None)
```

Dabei kann der Parameter **projection** folgende Werte haben:

```
cea Cylindrical Equal Area  
mbtfpq McBryde-Thomas Flat-Polar Quartic  
aeqd Azimuthal Equidistant  
sinu Sinusoidal  
poly Polyconic  
omerc Oblique Mercator  
gnom Gnomonic  
moll Mollweide  
lcc Lambert Conformal  
tmerc Transverse Mercator  
nplaea North-Polar Lambert Azimuthal  
gall Gall Stereographic Cylindrical  
npaeqd North-Polar Azimuthal Equidistant  
mill Miller Cylindrical  
merc Mercator  
stere Stereographic  
eqdc Equidistant Conic  
rotpole Rotated Pole  
cyl Cylindrical Equidistant  
npstere North-Polar Stereographic  
spstere South-Polar Stereographic  
hammer Hammer  
geos Geostationary  
nsper Near-Sided Perspective  
eck4 Eckert IV  
aea Albers Equal Area  
kav7 Kavrayskiy VII  
spaeqd South-Polar Azimuthal Equidistant  
npaepd North-Polar Azimuthal Equidistant  
ortho Orthographic  
cass Cassini-Soldner  
vandg van der Grinten  
laea Lambert Azimuthal Equal Area  
splaea South-Polar Lambert Azimuthal  
nplaea North-Polar Lambert Azimuthal  
robin Robinson
```

Die Parameter `width`, `height` geben die Breite, beziehungsweise die Höhe der Karte in Metern an, diese Parameter können bei den folgenden Projektionen nicht gesetzt werden.

`sinu`, `moll`, `hammer`, `npstere`, `spstere`, `nplaea`, `splaea`, `npaepd`, `spaepd`,
`robin`, `eck4`, `kav7`, `mbtfpq`, `ortho`, `geos`, `nsper`

Die Parameter `lon_0`, `lat_0` nehmen die Koordinate des Kartenmittelpunkts in Grad, beziehungsweise den zentralen Längen- oder Breitengrad.

Die Parameter `urcrnlon`, `urcrnlat`, `llcrnlon`, `llcrnlat`, `urcrnrx`, `urcrnry`, `llcrnrx`, `llcrnry` sind die Koordinaten der unteren linken und oberen rechten Ecke der Karte, in Grad oder Meter mit (0,0) im Mittelpunkt der Karte. Eine der beiden Arten die Grenzen der Karte festzulegen muss angegeben werden außer bei den folgenden Projektionen:

`sinu`, `moll`, `hammer`, `npstere`, `spstere`, `nplaea`, `splaea`, `npaepd`, `spaepd`,
`robin`, `eck4`, `kav7`, `mbtfpq`

Da diese entweder immer den ganzen Globus zeigen oder die Grenzen automatisch ermittelt werden. Bei der `rotpole` Projektion werden mit `lat/lon` die Ecken des nicht rotierten Globus angegeben mit `x/y` die Ecken des rotierten Globus. Bei dem Parameter `resolution` kann man die Werte `c`, `l`, `i`, `h`, `f`, `None` haben, falls `None` angegeben wurde, werden keine Grenzdaten geladen weshalb Klassenmethoden Fehler werfen.

Der Parameter `area_tresh` gibt an welche Mindest-Fläche Seen haben müssen um gezeichnet zu werden. Der Defaultwert ist 10000, 1000, 100, 10, 1 für die Resolution von `c`, `l`, `i`, `h`, `f`. Die Werte geben die Fläche in km^2 .

Der Parameter `rsphere` bekommt den Radius, der der Projektion zugrunde liegenden Figur, in Metern. Bei einer Kugel wird nur ein Wert angegeben, bei einem Ellipsoiden wird ein Array mit zwei Werten erwartet.

Über den Parameter `ellps` kann man die Form der Erde mit Hilfe spezieller Zeichenketten angeben. Sollte `ellps` angegeben sei wird `rsphere` ignoriert.

Mit dem Parameter `supress_ticks` kann man steuern ob die Achsen automatisch beschriftet und aufgeteilt werden sollen.

Der Parameter `fix_aspect` passt die Seitenverhältnisse vom Plot den Seitenverhältnissen der Karte an.

Der Parameter `anchor` gibt an wo die Karte im Gitter liegt, gültige Werte für diesen Parameter sind:

`C`, `SW`, `S`, `SE`, `E`, `NE`, `N`, `NW`, `W`

Mit dem Parameter `celestial` kann man einstellen das eine astronomische Konvention bezüglich der Längengrade benutzt wird. Was bedeutet, dass negative Längengrade östlich des Nullmeridian liegen. Diese Einstellung impliziert das die Resolution auf `None` gesetzt ist.

Mit dem `ax` Parameter kann man eine eigene Achseninstanz übergeben, wenn nichts übergeben wird, versucht `basemap` sich die aktuelle Achseninstanz mit Hilfe von `pyplot.gca()` zu holen. Falls man nicht `pyplot` importiert kann man auch jeder Klassenmethode die zeichnet eine Achseninstanz übergeben auf der gezeichnet wird. Wenn man allerdings hier die Achseninstanz übergibt werden alle Methoden auf dieser Achseninstanz ausgeführt.

Mit dem `lat_ts` Parameter wird der Breitengrad angegeben, der Maßstabs getreu wiedergegeben wird. Dieser Parameter ist optional und nur für die Projektionen `stere`, `cea`, `merc` von Belang.

Mit den Parametern `lat_1`, `lat_2` werden die Breitengrade der 1. und 2. Standardparallele angegeben. Diese sind nur für die Projektionen `lcc`, `aea`, `eqdc` interessant.

Mit den Parametern `lon_1`, `lon_2` werden die Breitengerade für die Punkte auf der Zentralen Linie der `omerc` Projektion angegeben.

Der Parameter `k_0` gibt den Skalierungsfaktor am Ursprung an, welcher nur von den Projektionen `tmerc`, `omerc`, `stere`, `lcc` genutzt wird.

Mit dem Parameter `no_rot` kann man bei der `omerc` Projektion einstellen ob die Koordinaten rotiert werden oder nicht. Die Parameter `o_lat_p`, `o_lon_p` geben die Position des rotierten Pols in Grad an, dies wird nur von der Projektion `rotpole` verarbeitet.

Der Parameter `boundinglat` gibt an bis zu welchem Breitengrad die polare Karte reicht.

Mit dem Parameter `round` kann man einstellen ob an dem Grenzbreitengrad bei polaren Karten abgeschnitten werden soll oder nicht. Dadurch kann man einstellen ob die Karte rund oder eckig ist.

Mit dem Parameter `satellite_height` gibt man die Höhe des Satelliten über dem Äquator an. Dies ist nur für die Projektionen `geos`, `nsper` interessant.

Mit dieser Funktion kann man eine Karteninstanz erstellen, allerdings ist noch nichts gezeichnet worden. Dafür müssen noch weitere Funktionen aufgerufen werden. Welche ich im nächsten Unterkapitel beschreibe.

3.4 Zeichnen der Karte

Die Karte zeichnet man mit verschiedenen Methoden der Karte.

Küstenlinie zeichnen

Die Küstenlinie zeichnet man mit der Funktion `drawcoastlines(linewidth=1.0, linestyle='solid', color='k', antialiased=1, ax=None, zorder=None)`, man kann mit den Parametern einstellen wie die Küstenlinie gezeichnet werden soll. Mit dem Parameter `zorder` kann man die Reihenfolge in der gezeichnet wird beeinflussen. Niedrige Werte im Parameter `zorder` werden zuerst gezeichnet. Mit dem Parameter `ax` kann man eine Achseninstanz übergeben auf

der gezeichnet werden soll, normalerweise wird die Achseninstanz der Karte genommen. Sollte keine Achseninstanz gefunden werden wird ein Fehler geworfen.

Ländergrenzen zeichnen

Ländergrenzen können mit der Funktion `drawcountries(linewidth=0.5, linestyle='solid', color='k', antialiased=1, ax=None, zorder=None)` gezeichnet werden. Die Parameter sind die selben wie beim zeichnen der Küstenlinien. Mit der Funktion `drawcounties(linewidth=0.1, linestyle='solid', color='k', antialiased=1, facecolor='none', ax=None, zorder=None, drawbounds=False)` können die Countiegrenzen in den USA gezeichnet werden. Mit dem Parameter `facecolor` kann eine Farbe angegeben werden mit der gefüllt werden soll. Mit der Funktion `drawstates(linewidth=0.5, linestyle='solid', color='k', antialiased=1, ax=None, zorder=None)` können die Staatsgrenzen in den USA gezeichnet werden.

Fülle der Karte

Mit der Funktion `fillcontinents(color='0.8', lake_color=None, ax=None, zorder=None, alpha=None)` kann man die Kontinente und Seen mit Farben füllen. Mit der Funktion `drawmapboundary(color='k', linewidth=1.0, fill_color=None, zorder=None, ax=None)` kann man die Kartengrenze zeichnen und die Karte mit einer Farbe füllen. So kann man die Ozeane mit einer Farbe füllen, wenn man die Kontinente ebenfalls mit Farbe füllt. Sonst hat alles die gleiche Farbe.

Man kann die Karte auch mit Hilfe der Funktion `drawlsmask(land_color='0.8', ocean_color='w', lsmask=None, lsmask_lons=None, lsmask_lats=None, lakes=True, resolution='l', grid=5, **kwargs)` füllen. Diese Funktion zeichnet ein Bild, daher kann man keine Zeichenreihenfolge angeben. Die Funktion erwartet im Parameter `lsmask` eine Matrix mit den Werten 0 für ein Ozean Pixel, 1 für ein Land Pixel und 2 für ein See Pixel. In den Parametern `lsmask_lon` und `lsmask_lat` erwartet die Funktion eindimensionale Arrays für die Längen- und Breitengrade der `lsmask`, sie müssen aufsteigend sortiert sein.

Flüsse zeichnen

Die Funktion `drawrivers(linewidth=0.5, linestyle='solid', color='k', antialiased=1, ax=None, zorder=None)` ermöglicht es Flüsse zu zeichnen. Es sind allerdings nicht alle Flüsse erfasst. Die größeren sind allerdings erfasst.

Längen- und Breitengrade zeichnen

Längen- und Breitengrade kann man einfach mit den Funktionen `drawmeridians(meridians, color='k', linewidth=1.0, zorder=None, dashes=[1, 1], labels=[0, 0, 0, 0], labelstyle=None, fmt='%g', xoffset=None, yoffset=None, ax=None, latmax=None, **kwargs)` und `drawparallels(circles, color='k', linewidth=1.0, zorder=None, dashes=[1, 1], labels=[0, 0, 0, 0], labelstyle=None, fmt='%g', xoffset=None, yoffset=None, ax=None, latmax=None, **kwargs)` zeichnen. Der Parameter `labels` gibt an wo die Grade beschriftet werden sollen links, rechts, oben, unten. Mit dem Parameter `labelstyle` kann man einstellen ob die Grade mit \pm oder mit E, W beziehungsweise N, S beschriftet werden sollen. Falls nichts angegeben wird, werden die Buchstaben benutzt. Dem Parameter `fmt` kann eine Funktion übergeben werden die aus dem Gradwert eine Zeichenkette macht. Mit dem Parameter `meridians` beziehungsweise `circles` wird eine Liste der Grade übergeben die gezeichnet werden sollen. Über den Parameter `dashes` kann man ein Pattern definieren mit dem die Grade gezeichnet werden sollen. Es werden immer abwechselnd die Anzahl an Pixeln angegeben, die gezeichnet und nicht gezeichnet werden sollen.

Maßstab zeichnen

Um einen Maßstab zu zeichnen braucht man die Funktion `drawmapscale(lon, lat, lon0, lat0, length, barstyle='simple', units='km', fontsize=9, yoffset=None, labelstyle='simple', fontcolor='k', fillcolor1='w', fillcolor2='k', ax=None, format='%d', zorder=None)`. Die Funktion zeichnet eine Skala an der Position `lon, lat` der Länge `length` von dem Punkt `lon0, lat0`. Die beiden Positionen sind wichtig da die Projektion längenverzerrend sein kann. Man kann den `barstyle` auf `fanzy` stellen, dann wird eine Skala gezeichnet bei der sich verschieden farbige Balken abwechseln. Wenn beim Parameter `labelstyle` `fanzy` angegeben wird, wird über dem Balken noch der Verzerrungsfaktor und die Position `lon0, lat0` ausgegeben. Mit dem Parameter `yoffset` kann man die Höhe der Skala und die Entfernung der Beschriftung vom Balken in Metern angeben. Der Defaultwert hierfür ist 2% der Karten Höhe.

Relief zeichnen

Um eine Karte mit Relief zu zeichnen kann man die Funktionen `etopo(ax=None, scale=None, **kwargs)` oder `shadedrelief(ax=None, scale=None, **kwargs)` benutzen. Die Funktion `etopo` lädt ein ein Reliefbild von der Seite <http://www.ngdc.noaa.gov/mgg/glob> als Hintergrund. Die Funktion `shadedrelief` lädt ein Reliefbild von der Seite <http://www.shadedrelief.com> als Hintergrund. Mit dem Parameter `scale` kann man einen Skalierungsfaktor angeben, da die Bilder 10800x5400 groß sind.

Was eine gewisse Zeit zum laden und verarbeiten braucht und Speicherplatz verbraucht.

3.5 Bild auf eine Karte zeichnen

Um ein Bild auf eine Karte zeichnen zu können, muss das Bild als `pyplot.image` vorliegen. Dies kann man mit Hilfe der PIL erreichen. Da `matplotlib.image` eine Funktion `pil_to_array(Pilimage)` zur Verfügung stellt. Beim Zeichnen von Bildern muss man beachten das sie über die ganze Karte gezeichnet werden. Die Funktion um ein Bild zu zeichnen ist `imshow(X, cmap=None, norm=None, aspect=None, interpolation=None, alpha=None, vmin=None, vmax=None, origin=None, extent=None, shape=None, filternorm=1, filterrad=4.0, imlim=None, resample=None, url=None, hold=None, **kwargs)`. Sie bekommt in `X` ein Bild als Pixelmatrix übergeben. Die anderen Parameter werden einfach an `pyplot.imshow()` weitergegeben. Die Parameter `extent` und `origin` werden automatisch so gesetzt, das das Bild über die ganze Karte gemalt wird.

Mit der Funktion `warpimage(image='bluemarble', scale=None, **kwargs)` kann man ein Hintergrundbild laden. Dieses Bild muss allerdings den ganzen Globus abdecken. Im Parameter `image` kann man einen Filenamen oder eine URL angeben. Sollte eine URL angegeben sein wird das Bild von der entsprechenden Seite als temporäre Datei herunter geladen. Dabei muss die URL mit `http` anfangen. Sollte nichts angegeben sein wird ein `blue marble next generation` Bild von `http://visibleearth.nasa.gov/` genommen.

Mit der Funktion `wmsimage(server, xpixels=400, ypixels=None, format='png', verbose=False, **kwargs)` kann man ein Hintergrundbild von einem WMS Server laden und zeichnen. Damit diese Funktion funktioniert muss die Karte mit dem passenden Parameter `epsg` erstellt worden sein, oder die Projektion `cyl` gewählt sein.

Mit der Funktion `arcgisimage(server='http://server.arcgisonline.com/ArcGIS', service='ESRI_Imagery_World_2D', xpixels=400, ypixels=None, dpi=96, verbose=False, **kwargs)` kann man ein Hintergrundbild von einem ArcGIS Server laden und darstellen. Mit dem Parameter `service` kann man einstellen welcher Art das Bild sein soll. Um diese Funktion benutzen zu können muss beim erstellen der Karte der Parameter `epsg` passend gesetzt worden sein, oder die Projektion `cyl` gewählt sein. Mit den Parametern `xpixels` und `ypixels` kann man die Anzahl Pixel in der Breite und Höhe einstellen. Sollte `ypixels` nicht gesetzt sein wird die Anzahl Pixel in der Höhe von der Anzahl Pixel in der Breite berechnet, so dass das Seitenverhältnis beibehalten bleibt.

Mit der Funktion `bluemarble(ax=None, scale=None, **kwargs)` kann man ein `blue marbel` Bild als Hintergrund malen. Mit Hilfe des Parameters `scale` kann man die Pixel Anzahl reduzieren, damit das Bild schneller geladen werden kann. Die Standardgröße des Bildes ist 5400x2700.

3.6 Daten plotten

Das Module **Basemap** bietet die Möglichkeit die Plotroutinen von **pyplot** zu benutzen, da es die Karte mit **matplotlib** erstellt. Die Karte ist im Grunde ein **AxesImage**, auf dem dann die entsprechenden Operationen ausgeführt werden. Daher kann man auch die Plotroutinen von **pyplot** aufrufen da diese ebenfalls mit **AxesImages** arbeiten. Wenn man mit **pyplot** zeichnet muss man beachten, dass die Koordinaten, die man angeben muss, in den Projektionskoordinaten der Karte anzugeben sind. Dabei ist die **Basemap** Klasse sehr hilfreich, sie rechnet nämlich Koordinaten in Längen- und Breitengrad in die Projektionskoordinaten um. Dazu muss man einfach die **basemap** Instanz mit den Koordinaten als Parameter aufrufen. Dabei können auch mehrere Koordinaten übergeben werden, indem man zwei Felder übergibt.

3.6.1 Isobaren plotten

um Isobaren zu plotten braucht man ein zweidimensionales Datenfeld mit den Werten und zwei Felder mit den dazu gehörigen Koordinaten. Diese kann man dann einfach mit Hilfe der Funktion `contour(x, y, data, *args, **kwargs)` plotten. Die Funktion `contour` zeichnet Isobarenlinien. Mit der Funktion `contourf(x, y, data, *args, **kwargs)` werden diese auch gefüllt. Den beiden Funktionen kann man ein `colormap` Objekt im Parameter `cmap` übergeben. Dieser Parameter wird dann einfach an **pyplot** weitergegeben. Darüber kann man den Farbverlauf der Isobaren steuern. Über den Parameter `levels` kann man angeben welche Wertelevel gezeichnet werden sollen.

3.6.2 Windmarken plotten

Mit der Funktion `barbs(**kw)` kann man einfach Windmarken plotten. Dazu übergibt man der Funktion die Koordinaten der Marke und die Koordinaten des Endpunkts des Vektors der dargestellt werden soll. Die Form der Marke wird durch die Länge des Vektors bestimmt. Eine Flagge bedeutet einen Wert von 50, ein voller Balken 10, ein halber Balken 5. Mit dem Parameter `barb_increments` kann man eigene Werte festlegen. Dafür wird dem Parameter ein dictionary mit den Schlüsseln `half`, `full`, `flag` übergeben. Mit dem Parameter `length` kann man die Länge der Fähnchen in Punkten angeben, der Rest der Marke wird dagegen skaliert.

3.6.3 Windvektoren plotten

Mit der Funktion `quiver(**kw)` kann man Vektoren plotten. Hierbei werden wieder 2 Koordinaten angegeben, wie bei dem Plotten von Windmarken.

Mit den Parametern `u,v` wird ein Vektor übergeben der gezeichnet werden soll. Mit den Parametern `scale`, `scale.units` kann man bestimmen wie lang die Vektoren werden. Je kleiner der `scale` Parameter desto größer wird der Vektor. Wenn man bei diesen Parametern nichts angibt wird der Wert automatisch aus den zu zeichnenden Vektoren ermittelt.

3.6.4 gekrümmte Strecken plotten

Um Strecken wie zum Beispiel eine Flugroute von New York nach London zu plotten, benutzt man am Einfachsten die Funktion `drawgreatcircle(lon1, lat1, lon2, lat2, del_s=100.0, **kwargs)`. Diese Funktion zeichnet einen Bogen von den Koordinaten `lon1,lat1` nach `lon2,lat2` dabei werden Punkte alle `del_s` km ermittelt. Diese Punkte kann man sich auch mit der Funktion `gcpoints(lon1, lat1, lon2, lat2, points)` berechnen lassen.

3.6.5 Viele Punkte plotten

Mit der Funktion `scatter(x, y, s=20, c=u'b', marker=u'o', cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, verts=None, hold=None, **kwargs)` kann man Marken plotten. Diese werden an die Positionen die durch `x,y` definiert sind gezeichnet. der Parameter `s` gibt die Größe der Marke an. Der Parameter `marker` gibt an was für eine Marke gezeichnet werden soll. Über den Parameter `c` kann man die Farben der Marken bestimmen.

3.6.6 Linienzüge zeichnen

Linienzüge lassen sich mit der Funktion `plot(x,y,**kw)` zeichnen. Diese Funktion zeichnet eine Linie von einem Punkt zum Nächsten. Dabei kann man `x,y` einfach als Felder übergeben. Hierbei ist zu beachten das davon ausgegangen wird das die Koordinaten Projektionskoordinaten sind.

3.7 Karte speichern

Um die Karte die man erstellt hat zu speichern bedient man sich des Moduls `pyplot`. Von diesem Modul kann man die Funktion `savefig(fname, dpi=None, facecolor='w', edgecolor='w', orientation='portrait', papertype=None, format=None, transparent=False, bbox_inches=None, pad_inches=0.1, frameon=None)` benutzen um die Figur zu speichern. Dazu gehören auch die Farblegende und der Titel der Figur. Es wird also nicht nur die Karte gespeichert. Mit dem Parameter `fname` wird der Dateiname angegeben in den gespeichert werden soll. Mit dem Parameter `format` kann ein unterstütztes Format angegeben werden, diese können variieren, meistens werden die Formate `png`, `pdf`, `ps`,

`eps`, `svg` unterstützt. Die Parameter `papertype` und `orientation` werden eventuell nur für das `ps` Format unterstützt.

3.8 Karte anzeigen

Mit der `pyplot` Funktion `show()` kann man die Karte anzeigen lassen. In der Anzeige kann man dann auch zoomen und die Karte verschieben. Was es einem ermöglicht auch Zeichnungen außerhalb der eigentlichen Karte zu sehen.

Literatur

<http://matplotlib.org/basemap> Basemap Dokumentation.

Nomenklatur

\mathcal{R}

Der Radius der Erde.

φ

Der Breitengrad der Polarkoordinate.

λ

Der Längengrad der Polarkoordinate.

\mathcal{X}

Die X Koordinate der 2D Projektion.

\mathcal{Y}

Die Y Koordinate der 2D Projektion.

Glossar

Flächentreu

Flächentreu heißt, dass der Maßstab mit dem Flächen verkleinert werden auf der gesamten Karte gleich ist. Dies führt an den Rändern zu Verzerrungen.

Winkeltreu

Winkeltreu heißt, dass die Winkel beziehungsweise die Richtungen von geraden beibehalten bleiben.

Längentreu

Längentreu heißt, dass die Entfernungen zwischen Punkten mit dem gleichen Maßstab verkleinert werden.